Mini SQL Project: Railway Ticket Management

System

1. Project Overview

This project manages railway ticket bookings, cancellations, passenger details, train information, and

analytical reports using SQL.

2. Database Tables

• Passengers (p_id, p_name)

• Tickets (t_id, p_id, source, destination, ticket_date, status)

• Train (t_id, train_name, routes)

3. Sample SQL Queries Implemented

Daily Ticket Report

SELECT ticket_date, COUNT(*) AS total_tickets, SUM(CASE WHEN status='booked' THEN 1 ELSE 0

END) AS booked_tickets, SUM(CASE WHEN status='cancel' THEN 1 ELSE 0 END) AS cancel_tickets

FROM tickets GROUP BY ticket_date ORDER BY ticket_date;

Route-wise Passengers

SELECT source + '->' + destination AS route, p.p_name, t.status FROM tickets t JOIN passengers p

ON t.p_id = p.p_id;

Status-wise Ticket Count

SELECT status, COUNT(*) FROM tickets GROUP BY status;

Most Crowded Routes

SELECT source + '->' + destination AS route, COUNT(*) AS total_passengers FROM tickets GROUP

BY source, destination ORDER BY total_passengers DESC;

Passenger Travel History

```
SELECT p.p_name, source + '->' + destination AS route, ticket_date, status FROM passengers p JOIN
tickets t ON p.p_id = t.p_id;
```

Daily Cancellation Report

```
SELECT ticket_date, COUNT(*) FROM tickets WHERE status='cancel' GROUP BY ticket_date
ORDER BY ticket_date;
```

Daily Booked Tickets Report

```
SELECT ticket_date, COUNT(*) FROM tickets WHERE status='booked' GROUP BY ticket_date
ORDER BY ticket_date;
```

4. Conclusion

This SQL project demonstrates passenger management, ticket booking analytics, and route-based

reporting,